A language-modelling approach to User-Centred Health Information Retrieval

Suzan Verberne

Institute for Computing and Information Sciences
Radboud University Nijmegen
s.verberne@cs.ru.nl

Abstract. In this working notes paper we present our methodology and the results we obtained in task 3a of the CLEF eHealth lab 2014. In the set-up of our experiments we assumed that the discharge summary provides the *context* of the patient's query, and therefore may contain useful background information that can be used to retrieve more relevant results. The central component in our approach is the Indri search engine, with its Language Modelling-based retrieval model. We experimented with query expansion using terms extracted from the discharge summary, and using terms extracted from the UMLS thesaurus. We obtained a small positive effect from expanding the Indri queries with terms from both sources. Our future work is directed at improving our term extraction and query expansion strategies.

1 Introduction

In this working notes paper we present our methodology and the results we obtained in task 3a of the CLEF eHealth lab 2014[1]. The aim of the eHealth track is "to evaluate systems that support laypeople in searching for and understanding their health information" [1, 3]. The goal of task 3 is "to provide valuable and relevant documents to patients, so as to satisfy their health-related information needs" [2]. The situation that the task simulates is a patient who has learned from his physician what his diagnosis is, and then starts searching the internet for medical information about his illness. The physician's information about the patient has been registered in the patient's discharge summary.

The data that was distributed for task 3 consists of:

- the 2012 crawl of approximately one million medical documents made available by the EU-FP7 Khresmoi project[2] in plain text form;
- a set of English 'layman' queries that individuals may realistically pose based on the content of their discharge summaries. The query creators used the disorder diagnosed in the discharge summary as main query term. The training set contained 5 queries and the test set 50;
- (optionally, after following human subjects training and obtaining the certificate) the collection of 299 discharge summaries from task 2 among which the discharge summaries belonging to the train and test queries for task 3.

In the set-up of our experiments we assumed that the discharge summary provides the *context* of the patient's query, and therefore may contain useful background information that can be used to retrieve more relevant results. Our research question is:

[1] http://clefehealth2014.dcu.ie/task-3
[2] http://www.khresmoi.eu/

Does query expansion with terms from the medical context of a patient's query lead to better results?

The central component in our approach is the Indri search engine[3], with its Language Modelling-based retrieval model. We experiment with query expansion using terms extracted from the discharge summary, and using terms extracted from the UMLS thesaurus.

2 Our approach

2.1 Data preprocessing

Document collection We preprocessed the document collection by splitting the corpus files in separate documents and saving for each document its uid, date, url and content.

Discharge summaries We obtained the corpus of 299 discharge summaries that was distributed for CLEF-eHealth task 2. We processed all discharge summaries that are referred to by a query in the query set from task 3. We cleaned the discharge summaries from all variables of the form `[** ...**]` (e.g. `[**MD Number 2860**]`), from abbreviations (using the regular expression `[a-z]\.([a-z]\.)+`), and from numbered lists (sequences of lines matching the regular expression `^[0-9]+\..*$`).

2.2 Indexing and Retrieval

We used the Indri API to index the CLEF collection and set up a query interface to the index. We applied a stopword list to the CLEF collection at indexing time. As ranking model, we use the Indri LM with Dirichlet smoothing and Pseudo-Relevance feedback (PRF) using the Ponte Expander. As parameters for the PRF, we used: number of feedback docs: 20, number of feedback terms: 3. We did not optimize these parameters for the current task but used the optimal settings from a previous task.

2.3 Query construction

All characters that are not alphanumeric, no hyphen or whitespace are removed from the query and all letters are lowercased. The words in the query are concatenated into one string and combined using the `combine` function in the Indri query language. We used two types of queries: short queries, consisting of a concatenation of the title and description fields, and long queries, consisting of a concatenation of all four fields title, description, profile and narr. For example, for this query:

```
<id>QTRAIN2014.1</id>
<discharge_summary>08114-027513-DISCHARGE_SUMMARY.txt
</discharge_summary>
<title>MRSA and wound infection</title>
<desc>What is MRSA infection and is it dangerous?</desc>
<profile>This 60 year old lady has had coronary artery bypass
grafting surgery and during recovery her wound has been infected.
She wants to know how dangerous her infection is, where she got
it and if she can be infected again with it.</profile>
<narr>Documents should contain information about sternal wound
infection by MRSA. They should describe the causes and the
complications.</narr>
```

[3] http://www.lemurproject.org/indri/

Table 1. The top-5 informative terms that we extracted from the discharge summaries belonging to training queries 2, 3 and 4 using the method by [4]

QTRAIN2014.2	QTRAIN2014.3	QTRAIN2014.4
mcg	unit stay	aortic dissection
mute	medical intensive care	lake
thrush	end-stage renal disease	type
levothyroxine sodium	department by	nodule
mesylate	saturating	right kidney

the following short query is created: `#combine(mrsa and wound infection what is mrsa infection and is it dangerous)`

2.4 Query expansion with terms from discharge summaries

We used the discharge summaries to create more informative queries, covering some of the medical background information of the patient. To this end, we aimed to extract the most relevant terms from the discharge summary belonging to each query. We extracted all n-grams with $n = \{1, 2, 3\}$ from the discharge summary and ranked them by their relevance. For calculating the relevance of each n-gram, we used Kullback-Leibler divergence for informativeness and phraseness [4]. In this method, term relevance is based on the expected loss between two language models, measured with point-wise Kullback-Leibler divergence. Tomokiyo and Hurst propose to mix two models for term scoring:

- phraseness (how tight are the words in the multi-word term):

$$kldivP = P(t) * log \frac{P(t)}{\prod_{i=1}^{n}(P(u_i))} \quad (1)$$

  in which $P(u_i)$ is the probability of the ith unigram inside the n-gram t
- informativeness (how informative is the term for the foreground corpus):

$$kldivI = P(t)_{fg} * log \frac{P(t)_{fg}}{P(t)_{bg}} \quad (2)$$

  in which $P(t)_{fg}$ is the probability of the n-gram t in a foreground collection and $P(t)_{bg}$ is the probability of t in the background collection.

For expanding query q, we used the discharge summary belonging to q as foreground collection, and all 299 discharge summaries in the task2 corpus as background collection. The parameter γ is the weight of the informativeness score relative to the phraseness score:

$$TermRelevance = \gamma * kldivI + (1 - \gamma) * kldivP \quad (3)$$

We used $\gamma = 0.9$, giving a higher weight to informativeness than to phraseness. We sort the n-grams by their $TermRelevance$. Table 1 shows the top-5 terms extracted for a few training queries. Note that not all terms are qualitative, some seem too generic to give information about this specific patient. This is caused by the relative sparseness of the data from which the terms are extracted, only one document.

In the runs that use the discharge summaries (run2–4), we added the top-k ($k = \{2, 5\}$) of terms extracted from the discharge summary to the query, again using unstructured queries with the `#combine`-operator. This implies that multi-word terms are treated as separate words in the query, not as phrases.

```
C UMG6-MRSA-METHICILLIN-RESISTANT-STAPHYLOCOCCUS-AUREUS-INFECTION: *0 #{{{
   S *0
      MRSA - Methicillin resistant Staphylococcus aureus infection *0
      Methicillin-resistant staphylococcus aureus (MRSA) *0
   .
-#}}}

C UMG9-METHICILLIN-RESISTANT-STAPHYLOCOCCUS-AUREUS: *0 #{{{
   S *0
      Methicillin resistant Staphylococcus aureus *0
      methicillin resistant Staphylococcus aureus *0
      MRSA *0
   .
-#}}}
```

Fig. 1. Two thesaurus entries from UMLS for the query title "MRSA and wound infection"

2.5 Query expansion with terms from a thesaurus

We use thesaurus expansion to enricht the queries with non-personalized information. The idea is that the patient might use layman's terminology and that the technical-medical synonyms of these terms might give better retrieval results. For this purpose, we use an old version of the ULMS-thesaurus that we have stored locally. In the UMLS, medical terms are ordered in synonym sets. An example is shown in Figure 1. We looked up all query titles from the train and test set in the ULMS: First we preprocessed the query titles so that only the first content word is kept (e.g. "MRSA and wound infection" becomes "MRSA"). Then we processed all synonym sets that

- contain one of the preprocessed query titles, or
- in which a word from the query title that is between 3 and 5 uppercase characters (presumably an abbreviation) is used between brackets (e.g. Methicillin-resistant staphylococcus aureus (MRSA))

For each query title, we created one *expansion set* by merging all synonym sets in which the preprocessed query title occurs, disregarding synonym sets with 30 or more terms in them (because generic synonym sets such as "UMG2-BODY-PART-ORGAN-OR-ORGAN-COMPONENT" contain dozens and sometimes hundreds of terms). From the expansion set, we removed duplicates and near-duplicates (the only difference is a plural -s, hyphens and underscores, or capitalization). Then we sort the expansion terms by the number of times they occur in synonym sets together with the query title and select the top-k terms to be added to the query.

In the runs that use the thesaurus (run2,3,5,6), we added the top-k ($k = 5$) of terms extracted from the thesaurus to the query, again using unstructured queries with the `#combine`-operator. This implies that multi-word terms are treated as separate words in the query, not as phrases.

3 Submitted runs

1. Baseline: Indri LM retrieval with Pseudo-Relevance feedback (PRF). Queries are unstructured combination of title and description, using the `#combine`- operator in the Indri query language
2. Same as Run1, and:

Table 2. Evaluation of our runs in terms of Precision (P), nDCG, MAP and the number of relevant results retrieved (ret_rel). For each evaluation measure (column), the highest scoring run is marked in boldface.

Run ID	P@10	nDCG@10	MAP	ret_rel
NIJM_EN_Run.1	0.5740	0.5708	**0.3036**	**2330**
NIJM_EN_Run.2	**0.6180**	**0.6149**	0.2825	2190
NIJM_EN_Run.3	0.5960	0.5772	0.2606	2154
NIJM_EN_Run.4	0.5680	0.5669	0.2695	2176
NIJM_EN_Run.5	0.5880	0.5773	0.2609	2165
NIJM_EN_Run.6	0.5220	0.5302	0.2180	1939
NIJM_EN_Run.7	0.5220	0.5302	0.2180	1939

   - We expanded each query with terms from the discharge summary (k=5)
   - We expanded each query with maximally 5 terms from the UMLS thesaurus
3. Same as Run2, but with k=2 for the terms from the discharge summary
4. Same as Run2 (k=5), but without the thesaurus expansion. Note that in our original submission, there was an error in this run, as a result of which its results were equal to Run 3. We reproduced the run and included the correct results in the results section below.
5. Same as Run1, and:
   - We expanded each query with maximally 5 terms from the UMLS thesaurus
6. Same as Run5 but with long queries: unstructured combination of title, description, profile and narrative
7. Same as Run6 but without the thesaurus expansion

4 Results

4.1 Evaluation of the runs

Table 2 shows a summary of the results obtained for our runs. From the results, we make the following observations:

- Run 1, with short queries and no query expansion, gives the best results in terms of MAP and the number of retrieved and relevant results.
- Run 2, with 5 terms from the discharge summary and maximum 5 terms from the thesaurus, gives the best results in terms of nDCG@10 and P@10. This suggests that a combination of terms from the discharge summary and from the thesaurus added to the query can lead to better results in the top-10.
- Run 3, with 2 expansion terms from the discharge summary and maxumum 5 terms from the thesaurus, gives worse results than Run 2, with 5 terms from the discharge summary. This suggests that the most informative terms from the discharge summaries are not always ranked at top; or that a combination of terms is needed to retrieve more relevant results.
- Run 4, with 5 terms from the discharge summary and no terms from the thesaurus, does not beat the baseline Run 1. This suggests that without the thesaurus terms, the discharge summary terms do not lead to improvement.
- Run 5, with maximally 5 terms from the thesaurus and no terms from the discharge summary, gives almost the same results as Run 3, with 2 terms from the discharge summary. This suggests that if there is relevant information in the discharge summary terms, it only leads to improvement if we add enough terms (2 is not enough).
- Run 6, with long queries, no terms from the discharge summary and maximally 5 terms from the thesaurus, gives worse results than Run 5, with short queries and maximally 5 terms from the thesaurus. This suggests that short queries are better than long queries.

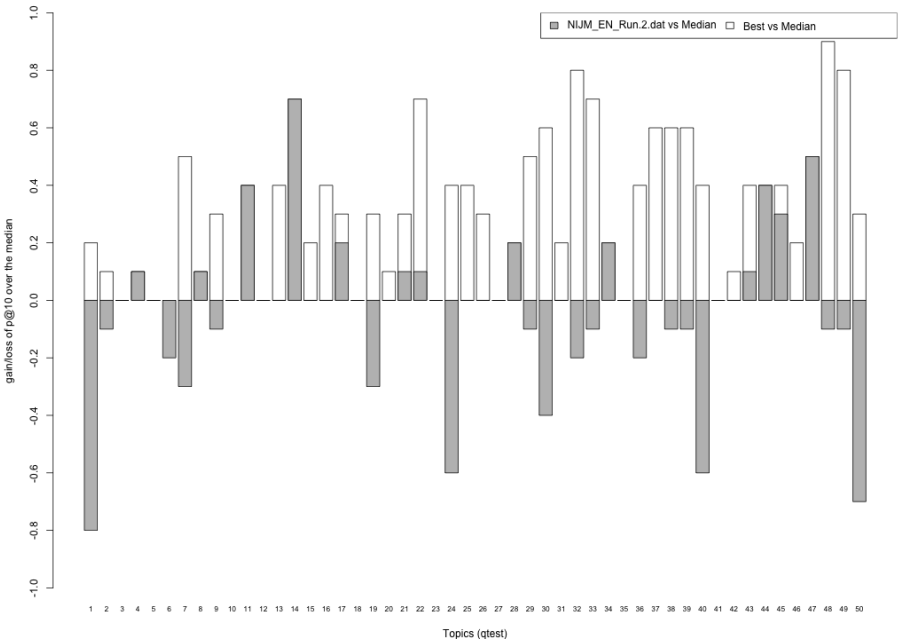

Fig. 2. Per-query results for Precision@10 for our best run (of the runs submitted), Run2. For each query, the height of a bar represents the gain/loss of our Run (grey) and the best Run (white) over the median run of all submitted runs.

– Run 7, with long queries and no thesaurus expansion gives the exact same results as Run 6, with long queries and maximally 5 terms from the thesaurus. This suggests that when using the long queries with the fields 'profile' and 'narrative' fully included, makes it senseless to add a few thesaurus terms. Since runs 6 and 7 are the poorest of all our runs, using the longer queries suspectedly leads to more irrelevant results than the shorter queries. Even if the thesaurus terms are informative, they cannot compensate this.

4.2 Per-query analysis

Figure 2 shows the per-query results for our best run (of the runs submitted), Run2. It shows a large divergence between queries. For some queries (the ones where there is no white bar), our run (the grey bar) scores the best of all runs, but for others, our run scores far below median. In follow-up work we will investigate the query characteristics that cause our method to be more or less successful. We suspect that the success depends at least partly on the quality of the extracted additional terms.

5 Conclusions

Since we did not do comprehensive analyses of our results, we can only draw preliminary conclusions. We worked with a Language Modelling approach, and although the Indri retrieval model can handle long queries (it will find the best possible matches for the combination all query terms — not necessarily all terms are present in the retrieved documents), we found that expanding the query consisting of the fields title and description with the fields profile and narrative has a negative effect on the retrieval results. However, we did get a positive effect from adding the 5 most informative terms from the discharge summary and maximally 5 synonyms from the thesaurus to the queries.

In the near future, we plan to do more analyses in order to find out what factors play a role in the success of query expansion. Our current line of work focuses on personalization of IR through terms extracted from personal documents. The discharge summary is a good example of a document that may provide additional information about the context of a user's query. For that purpose, we aim to improve our term extraction and query expansion strategies. In addition, we will investigate how thesaurus expansion could be applied successfully (selecting the query terms to look up, selecting the synonym sets to expand with).

Acknowledgements

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

References

1. Lorraine Goeuriot, G Jones, Liadh Kelly, Johannes Leveling, Allan Hanbury, Henning Müller, Sanna Salanterä, Hanna Suominen, and Guido Zuccon. Share/clef ehealth evaluation lab 2013, task 3: Information retrieval to address patients questions when reading clinical reports. *Online Working Notes of CLEF, CLEF*, 2013.
2. Lorraine Goeuriot, Liadh Kelly, Wei Li, Joao Palotti, Pavel Pecina, Guido Zuccon, Allan Hanbury, Gareth Jones, and Henning Mueller. Share/clef ehealth evaluation lab 2014, task 3: User-centred health information retrieval. In *Proceedings of CLEF 2014*, 2014.
3. Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Tobias Schrek, Gondy Leroy, Danielle L. Mowery, Sumithra Velupillai, Wendy W. Chapman, David Martinez, Guido Zuccon, and Joao Palotti. Overview of the share/clef ehealth evaluation lab 2014. In *Proceedings of CLEF 2014*, Lecture Notes in Computer Science (LNCS). Springer, 2014.
4. Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*, pages 33–40. Association for Computational Linguistics, 2003.